[illegible]
[illegible] 042

TRELLISES AND TRELLIS-BASED DECODING ALGORITHMS FOR LINEAR BLOCK CODES

Part 3

Shu Lin and Marc Fossorier

April 20, 1998

N165-931
IN-[illegible]

13 THE MAP AND RELATED DECODING ALGORITHMS

In a coded communication system with equiprobable signaling, MLD minimizes the word error probability and delivers the most likely codeword associated with the corresponding received sequence. This decoding has two drawbacks. First, minimization of the word error probability is not equivalent to minimization of the bit error probability. Therefore, MLD becomes suboptimum with respect to the bit error probability. Second, MLD delivers a hard-decision estimate of the received sequence, so that information is lost between the input and output of the ML decoder. This information is important in coded schemes where the decoded sequence is further processed, such as concatenated coding schemes, multi-stage and iterative decoding schemes.

In this chapter, we first present a decoding algorithm which both minimizes bit error probability, and provides the corresponding soft information at the output of the decoder. This algorithm is referred to as the MAP (maximum a-posteriori probability) decoding algorithm [1]. Unfortunately, the trellis-based implementation of the MAP algorithm is much more complex than that of the

trellis-based MLD algorithms presented in the previous chapters. Consequently, suboptimum versions of the MAP algorithm with reduced decoding complexity must be considered for many practical applications. In Section 13.2, we present a near-optimum modification of the MAP algorithm, referred to as the Max-Log-MAP (or SOVA) decoding algorithm [34, 39, 40]. This near-optimum algorithm performs within only few tenths of a dB of the optimum MAP decoding algorithm while reduces decoding complexity drastically. Finally, the minimization of the bit error probability in trellis-based MLD is discussed in the last section.

13.1 THE MAP DECODING ALGORITHM

Consider a binary (N, K) linear block code C. Let $u = (u_1, u_2, \ldots, u_N)$ be a codeword in C. Define $P(A|B)$ as the conditional probability of the event A given the occurrence of the event B. The MAP decoding algorithm evaluates the most likely bit value u_i at a given bit position i based on the received sequence $r = (r_1, r_2, \ldots, r_N)$. It first computes the log-likelihood ratio

$$L_i \triangleq \log \frac{P(u_i = 1|r)}{P(u_i = 0|r)} \tag{13.1}$$

for $1 \leq i \leq N$, and then compares this value to a zero-threshold to decode u_i as

$$u_i = \begin{cases} 1 & \text{for } L_i > 0, \\ 0 & \text{for } L_i \leq 0. \end{cases} \tag{13.2}$$

The value L_i represents the soft information associated with the decision on u_i. It can be used for further processing of the sequence u delivered by the MAP decoder.

In the N-section trellis diagram for the code, let $B_i(C)$ denote the set of all branches (σ_{i-1}, σ_i) that connect the states in the state space $\Sigma_{i-1}(C)$ at time-$(i-1)$ and the states in the state space $\Sigma_i(C)$ at time-i for $1 \leq i \leq N$. Let $B_i^0(C)$ and $B_i^1(C)$ denote the two disjoint subsets of $B_i(C)$ that correspond to the output code bits $u_i = 0$ and $u_i = 1$, respectively, given by (3.3). Clearly

$$B_i(C) = B_i^0(C) \cup B_i^1(C) \tag{13.3}$$

for $1 \leq i \leq N$. For $(\sigma', \sigma) \in B_i(C)$, we define the joint probability

$$\lambda_i(\sigma', \sigma) \triangleq P(\sigma_{i-1} = \sigma'; \sigma_i = \sigma; r) \tag{13.4}$$

for $1 \leq i \leq N$. Then

$$P(u_i = 0; \boldsymbol{r}) = \sum_{(\sigma',\sigma) \in B_i^0(C)} \lambda_i(\sigma', \sigma), \tag{13.5}$$

$$P(u_i = 1; \boldsymbol{r}) = \sum_{(\sigma',\sigma) \in B_i^1(C)} \lambda_i(\sigma', \sigma). \tag{13.6}$$

The MAP decoding algorithm computes the probabilities $\lambda_i(\sigma', \sigma)$ which are then used to evaluate $P(u_i = 0|\boldsymbol{r})$ and $P(u_i = 1|\boldsymbol{r})$ in (13.1) from (13.5) and (13.6).

For $1 \leq i \leq N$, $1 \leq l \leq m \leq N$, and $\boldsymbol{r}_l^m \triangleq (r_l, r_{l+1}, \ldots, r_m)$, we define the probabilities

$$\begin{aligned}
\alpha_i(\sigma) &\triangleq P(\sigma_i = \sigma; \boldsymbol{r}_1^i), && (13.7)\\
\beta_i(\sigma) &\triangleq P(\boldsymbol{r}_{i+1}^N | \sigma_i = \sigma), && (13.8)\\
\gamma_i(\sigma', \sigma) &\triangleq P(\sigma_i = \sigma; r_i | \sigma_{i-1} = \sigma') \\
&= P(r_i | (\sigma_{i-1}, \sigma_i) = (\sigma', \sigma))\, P(\sigma_i = \sigma | \sigma_{i-1} = \sigma'). && (13.9)
\end{aligned}$$

Then, for a memoryless channel,

$$\lambda_i(\sigma', \sigma) = \alpha_{i-1}(\sigma')\, \gamma_i(\sigma', \sigma)\, \beta_i(\sigma) \tag{13.10}$$

for $1 \leq i \leq N$, which shows that the values $\lambda_i(\sigma', \sigma)$ can be evaluated by computing all values $\alpha_i(\sigma)$, $\beta_i(\sigma)$ and $\gamma_i(\sigma', \sigma)$. Based on the total probability theorem, we can express $\alpha_i(\sigma)$, for $1 \leq i \leq N$, as follows:

$$\begin{aligned}
\alpha_i(\sigma) &= \sum_{\sigma' \in \Sigma_{i-1}(C)} P(\sigma_{i-1} = \sigma'; \sigma_i = \sigma; \boldsymbol{r}_1^{i-1}; r_i) \\
&= \sum_{\sigma' \in \Sigma_{i-1}(C)} \alpha_{i-1}(\sigma')\, \gamma_i(\sigma', \sigma). && (13.11)
\end{aligned}$$

Similarly, for $1 \leq i < N$,

$$\begin{aligned}
\beta_i(\sigma) &= \sum_{\sigma' \in \Sigma_{i+1}(C)} P(r_{i+1}; \boldsymbol{r}_{i+2}^N; \sigma_{i+1} = \sigma' | \sigma_i = \sigma) \\
&= \sum_{\sigma' \in \Sigma_{i+1}(C)} \beta_{i+1}(\sigma')\, \gamma_{i+1}(\sigma, \sigma'). && (13.12)
\end{aligned}$$

From (13.11), we see that the probabilities $\alpha_i(\sigma)$ with $1 \le i \le N$ can be computed recursively from the initial state σ_0 to the final state σ_f of the N-section trellis for code C, once $\gamma_i(\sigma',\sigma)$'s are computed. This is called the **forward recursion.** From (13.12), we see that the probabilities $\beta_i(\sigma)$ with $1 \le i \le N$ can be computed recursively in backward direction from the final state σ_f to the initial state σ_0 of the N-section trellis of C. This is called the **backward recursion.**

For the AWGN channel with BPSK transmission, we have

$$P(r_i|\sigma_{i-1} = \sigma'; \sigma_i = \sigma) = (\pi N_0)^{-1/2} \exp(-(r_i - c_i)^2/N_0)\, \delta_i(\sigma',\sigma), \quad (13.13)$$

where

$$\delta_i(\sigma',\sigma) = \begin{cases} 1 & \text{if } (\sigma',\sigma) \in B_i(C), \\ 0 & \text{if } (\sigma',\sigma) \notin B_i(C). \end{cases} \quad (13.14)$$

For $\delta_i(\sigma',\sigma) = 1$, $c_i = 2u_i - 1$ is the transmitted signal corresponding to the label u_i of the branch (σ',σ). Based on (13.13), $\gamma_i(\sigma',\sigma)$ is proportional to the value

$$w_i(\sigma',\sigma) = \exp(-(r_i - c_i)^2/N_0)\, \delta_i(\sigma',\sigma)\, P(\sigma_i = \sigma|\sigma_{i-1} = \sigma'). \quad (13.15)$$

Note that in many applications such as MAP trellis-based decoding of linear codes, the a-priori probability of each information bit is the same, so that all states $\sigma \in \Sigma_i(C)$ are equiprobable. Consequently, $P(\sigma_i = \sigma|\sigma_{i-1} = \sigma')$ becomes a constant that can be discarded in the definition of $w_i(\sigma',\sigma)$. However, this is not true in general. For example, in iterative or multi-stage decoding schemes $P(\sigma_i = \sigma|\sigma_{i-1} = \sigma')$ has to be evaluated after the first iteration, or after the first decoding stage. Since in (13.1), we are interested only in the ratio between $P(u_i = 1|r)$ and $P(u_i = 0|r)$, $\lambda_i(\sigma',\sigma)$ can be scaled by any value without modifying the decision on u_j. Based on these definitions, $\alpha_i(\sigma)$ can be computed recursively based on (13.11) using the trellis diagram from the initial state σ_0 to the final state σ_f as follows:

(1) Assume that $\alpha_{i-1}(\sigma')$ has been computed for all states $\sigma' \in \Sigma_{i-1}(C)$.

(2) In the i-th section of the trellis diagram, associate the weight $w_i(\sigma',\sigma)$ with each branch $(\sigma',\sigma) \in B_i(C)$.

(3) For each state $\sigma \in \Sigma_i(C)$, evaluate and store the weighted sum

$$\alpha_i(\sigma) = \sum_{\sigma' \in \Sigma_{i-1}(C): \delta_i(\sigma', \sigma)=1} w_i(\sigma', \sigma)\, \alpha_{i-1}(\sigma'). \qquad (13.16)$$

The initial conditions for this recursion are $\alpha_0(\sigma_0) = 1$ and $\alpha_0(\sigma) = 0$ for $\sigma \neq \sigma_0$.

Similarly, $\beta_i(\sigma)$ can be computed recursively based on (13.12) using the trellis diagram from the final state σ_f to the initial state σ_0 as follows:

(1) Assume that $\beta_{i+1}(\sigma')$ has been computed for all states $\sigma' \in \Sigma_{i+1}(C)$.

(2) In the $(i+1)$-th section of the trellis diagram, associate the weight $w_{i+1}(\sigma, \sigma')$ with each branch $(\sigma, \sigma') \in B_{i+1}(C)$.

(3) For each state $\sigma \in \Sigma_i(C)$, evaluate and store the weighted sum

$$\beta_i(\sigma) = \sum_{\sigma' \in \Sigma_{i+1}(C): \delta_{i+1}(\sigma, \sigma')=1} w_{i+1}(\sigma, \sigma')\, \beta_{i+1}(\sigma'). \qquad (13.17)$$

The corresponding initial conditions are $\beta_N(\sigma_f) = 1$ and $\beta_N(\sigma) = 0$ for $\sigma \neq \sigma_f$.

The MAP decoding algorithm requires one forward recursion from σ_0 to σ_f, and one backward recursion from σ_f to σ_0 to evaluate all values $\alpha_i(\sigma)$ and $\beta_i(\sigma)$ associated with all states $\sigma_i \in \Sigma_i(C)$, for $1 \leq i \leq N$. These two recursions are independent of each other. Therefore, the forward and backward recursions can be executed simultaneously in both directions along the trellis of the code C. This bidirectional decoding reduces the decoding delay. Once all values of $\alpha_i(\sigma)$ and $\beta_i(\sigma)$ for $1 \leq i \leq N$ have been determined, the values L_i in (13.1) can be computed from (13.5), (13.6), (13.9) and (13.10).

13.2 THE SOVA DECODING ALGORITHM

The MAP decoding algorithm presented in the previous section requires a large number of computations and a large storage to compute and store the probabilities $\alpha_i(\sigma)$, $\beta_i(\sigma)$ and $\gamma_i(\sigma', \sigma)$ for all the states σ and state pairs (σ', σ) in the trellis for the code to be decoded. For a long code with large trellis, the implementation of the MAP decoder is practically impossible. Also, the MAP decoding algorithm computes probability values, which require much more complicated real value operations than the real additions performed by trellis-based MLD algorithms, such as the Viterbi decoding and RMLD algorithms.

In this section, we present an algorithm for which the optimum bit error performance associated with the MAP algorithm is traded with a significant reduction in decoding complexity. This algorithm is known as the **Max-Log-MAP** algorithm or **soft-output Viterbi algorithm (SOVA)**, as it performs the same operations as the Viterbi algorithm, with additional real additions and storages.

For BPSK transmission, (13.1) can be rewritten as

$$L_i = \log\left[\left(\sum_{\mathbf{c}:c_i=1} P(\mathbf{c}|\boldsymbol{r})\right) / \left(\sum_{\mathbf{c}:c_i=-1} P(\mathbf{c}|\boldsymbol{r})\right)\right], \qquad (13.18)$$

for $1 \leq i \leq N$ and $c_i = 2u_i - 1$. Based on the approximation

$$\log(\sum_{j=1}^{N} \delta_j) \approx \log(\max_{j\in\{1,2,\ldots,N\}} \{\delta_j\}), \qquad (13.19)$$

we obtain from (13.18)

$$L_i \approx \log(\max_{\mathbf{c}:c_i=1} P(\mathbf{c}|\boldsymbol{r})) - \log(\max_{\mathbf{c}:c_i=-1} P(\mathbf{c}|\boldsymbol{r})). \qquad (13.20)$$

For each code bit u_i, the Max-Log-MAP algorithm [40] approximates the corresponding log-likelihood ratio L_i based on (13.20).

For the AWGN channel with BPSK transmission, we have

$$\begin{aligned} P(\mathbf{c}|\boldsymbol{r}) &= P(\boldsymbol{r}|\mathbf{c})P(\mathbf{c})/P(\boldsymbol{r}) \\ &= (\pi N_0)^{-N/2} e^{-\sum_{j=1}^{N}(r_j-c_j)^2/N_0} P(\mathbf{c})/P(\boldsymbol{r}). \end{aligned} \qquad (13.21)$$

If $\mathbf{c}^1 = (c_1^1, c_2^1, \ldots, c_{2j-1}^1, c_{2j}^1, \ldots)$ and $\mathbf{c}^0 = (c_1^0, c_2^0, \ldots, c_{2j-1}^0, c_{2j}^0, \ldots)$ represent the codewords corresponding to the first term and the second term of (13.20), respectively, it follows from (13.21) that for equiprobable signaling, the approximation of L_i given in (13.20) is proportional to the value

$$r_i + \sum_{j:j\neq i, c_j^0 \neq c_j^1} c_j^1 r_j. \qquad (13.22)$$

We observe that one of the terms in (13.20) corresponds to the MLD solution, while the other term corresponds to the most likely codeword which differs from the MLD solution in $\mathbf{u}_i$. Consequently, in Max-Log-MAP or SOVA decoding, the hard-decision codeword corresponding to (13.20) is the MLD codeword and (13.22) is proportional to the difference of squared Euclidean distances (SED) $\|\boldsymbol{r} - \mathbf{c}^1\|^2 - \|\boldsymbol{r} - \mathbf{c}^0\|^2$. For any two codewords $\mathbf{c}$ and $\mathbf{c}'$, we define $\left|\|\boldsymbol{r} - \mathbf{c}\|^2 - \|\boldsymbol{r} - \mathbf{c}'\|^2\right|$ as the reliability difference between $\mathbf{c}$ and $\mathbf{c}'$.

For simplicity, we consider the trellis diagram of a rate-1/2 antipodal convolutional code C. Hence, the two branches that merge into each state have different branch labels, as described in Figure 10.1. Also, we assume that the trellis diagram for the code C is terminated so that N encoded bits are transmitted. Generalization of the derived results to other trellis diagrams is straightforward, after proper modification of the notations. At each state σ_{i-1} of the state space $\Sigma_{i-1}(C)$ at time-$(i-1)$, the SOVA stores the cumulative correlation metric value $M(\sigma_{i-1})$ and the corresponding decoded sequence

$$\hat{c}(\sigma_{i-1}) = \left(\hat{c}_1(\sigma_{i-1}), \hat{c}_2(\sigma_{i-1}), \ldots, \hat{c}_{2(i-1)-1}(\sigma_{i-1}), \hat{c}_{2(i-1)}(\sigma_{i-1})\right), \quad (13.23)$$

as for the Viterbi algorithm. In addition, it also stores the reliability measures

$$\hat{L}(\sigma_{i-1}) = \left(\hat{L}_1(\sigma_{i-1}), \hat{L}_2(\sigma_{i-1}), \ldots, \hat{L}_{2(i-1)-1}(\sigma_{i-1}), \hat{L}_{2(i-1)}(\sigma_{i-1})\right), \quad (13.24)$$

associated with the corresponding decision $\hat{c}(\sigma_{i-1})$.

At the decoding time-i, for each state σ_i in the state space $\Sigma_i(C)$, the SOVA first evaluates the two cumulative correlation metric candidates $M(\sigma_{i-1}^1, \sigma_i)$ and $M(\sigma_{i-1}^2, \sigma_i)$ corresponding to the two paths terminating in state σ_i with transitions from states σ_{i-1}^1 and σ_{i-1}^2, respectively As for the Viterbi algorithm, the SOVA selects the cumulative correlation metric

$$M(\sigma_i) = \max_{l\in\{1,2\}} \{M(\sigma_{i-1}^l, \sigma_i)\}, \quad (13.25)$$

and updates the corresponding pair $(\hat{c}_{2i-1}(\sigma_i), \hat{c}_{2i}(\sigma_i))$ in the surviving path $\hat{c}(\sigma_i)$ at state σ_i. Next, $\hat{L}(\sigma_i)$ has to be updated To this end, we define

$$\Delta_i \triangleq \max_{l\in\{1,2\}} \{M(\sigma_{i-1}^l, \sigma_i)\} - \min_{l\in\{1,2\}} \{M(\sigma_{i-1}^l, \sigma_i)\}. \quad (13.26)$$

Based on (13.22) and the fact that the code considered is antipodal, we set

$$\hat{L}_{2i-1}(\sigma_i) = \hat{L}_{2i}(\sigma_i) = \Delta_i, \quad (13.27)$$

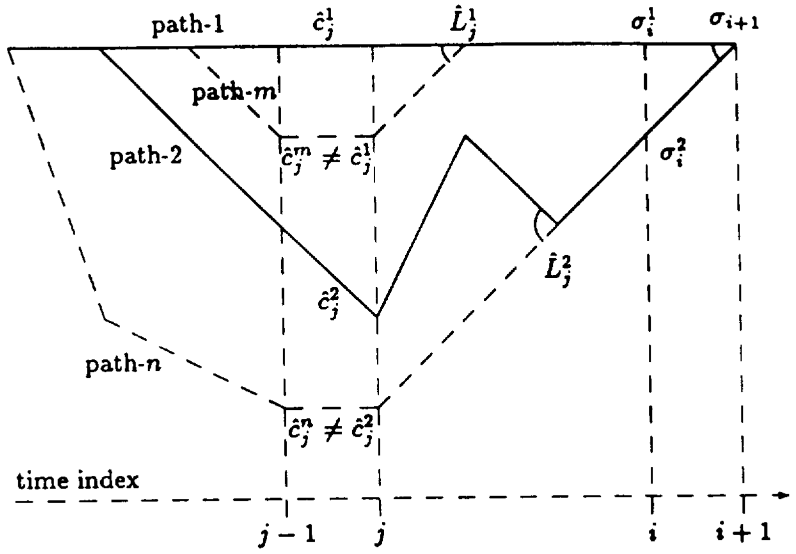

Figure 13.1. Trellis description with respect to reliability values associated with bit-j at state σ_{i+1}.

since Δ_i represents the reliability difference between the two most likely code-sequences terminating at state σ_i with different values for both $\hat{c}_{2i-1}$ and $\hat{c}_{2i}$. The remaining values $\hat{L}_j(\sigma_i)$ for $j = 1, \ldots, 2(i-1)$ of the surviving $\hat{L}(\sigma_i)$ at state σ_i have to be updated.

In the following, we simplify the above notations and define

$$\hat{L}(\sigma_{i-1}^l) = (\hat{L}_1^l, \hat{L}_2^l, \ldots, \hat{L}_{2(i-1)-1}^l, \hat{L}_{2(i-1)}^l) \tag{13.28}$$

for $l = 1, 2$, as the two sets of reliability measures corresponding to the two candidate paths merging into state σ_i with transitions from states σ_{i-1}^1 and σ_{i-1}^2, respectively. We refer to these two paths as path-1 and path-2, and without loss of generality assume that path-1 is the surviving path. Similarly, for $l = 1, 2$,

$$\hat{u}(\sigma_{i-1}^l) = (\hat{c}_1^l, \hat{c}_2^l, \ldots, \hat{c}_{2(i-1)-1}^l, \hat{c}_{2(i-1)}^l) \tag{13.29}$$

represent the two sets of decisions corresponding to path-1 and path-2, respectively.

First, we consider the case $\hat{c}_j^1 \neq \hat{c}_j^2$, for some $j \in \{1, \ldots, 2(i-1)\}$, and recall that path-1 and path-2 have a reliability difference equal to Δ_i. Also,

$\hat{L}_j^1$ represents the reliability difference between path-1 and a code-sequence represented by a path-m merging with path-1 between the decoding steps j and $(i-1)$, with $\hat{c}_j^m \neq \hat{c}_j^1$, as shown in Figure 13.1. On the other hand, $\hat{L}_j^2$ represents the reliability difference between path-2 and a code-sequence represented by a path-n merging with path-2 between the decoding steps j and $(i-1)$, with $\hat{c}_j^n = \hat{c}_j^1$. Hence, $\hat{L}_j^2$ does not need to be considered to update $\hat{L}_j(\sigma_i)$ in $\hat{L}(\sigma_i)$. Since no additional reliability information is available at state σ_i, we update

$$\hat{L}_j(\sigma_i) = \min\{\Delta_i, \hat{L}_j^1\}. \tag{13.30}$$

Next, we consider the case $\hat{c}_j^1 = \hat{c}_j^2$, for some $j \in \{1, \ldots, 2(i-1)\}$, so that path-2 is no longer considered to update $\hat{L}_j(\sigma_i)$ in $\hat{L}(\sigma_i)$. However, path-n previously defined now satisfies $\hat{c}_j^n \neq \hat{c}_j^1$. Since the reliability difference between path-1 and path-n is $\Delta_i + \hat{L}_j^2$ (i.e. the reliability difference between path-1 and path-2 plus the reliability difference between path-2 and path-n), we obtain

$$\hat{L}_j(\sigma_i) = \min\{\Delta_i + \hat{L}_j^2, \hat{L}_j^1\}. \tag{13.31}$$

The first version of SOVA that is equivalent to the above development was introduced by Battail in 1987 [3]. This algorithm was later reformulated in conjunction with the MAP algorithm in [9, 61] and formally shown to be equivalent to the Max-Log-MAP decoding algorithm in [34]. Consequently, the Max-Log-MAP or SOVA decoding algorithm can be summarized as follows.
For each state σ_i of the state space $\Sigma_i(C)$:

Step 1: Perform the Viterbi decoding algorithm to determine the survivor metric $M(\sigma_i)$ and the corresponding code-sequence $\hat{c}(\sigma_i)$.

Step 2: For $j \in \{1, \ldots, 2(i-1)\}$, set $\hat{L}_j(\sigma_i)$ in $\hat{L}(\sigma_i)$ either to the value $\min\{\Delta_i, \hat{L}_j^1\}$ if $\hat{c}_j^1 \neq \hat{c}_j^2$, or to the value $\min\{\Delta_i + \hat{L}_j^2, \hat{L}_j^1\}$ if $\hat{c}_j^1 = \hat{c}_j^2$.

Step 3: Set $\hat{L}_{2i-1}(\sigma_i)$ and $\hat{L}_{2i}(\sigma_i)$ in $\hat{L}(\sigma_i)$ to the value Δ_i.

In [39], a simplified version of SOVA is presented. It is proposed to update $\hat{L}_j(\sigma_i)$, for $j = 1, 2, \ldots, 2(i-1)$, only when $c_j^1 \neq c_j^2$. Hence (13.30) remains unchange while (13.31) simply becomes:

$$\hat{L}_j(\sigma_i) = \hat{L}_j^1. \tag{13.32}$$

Consequently, the values $\hat{L}_j^2$ are no longer needed in the updating rule, so that this simplified version of SOVA is easier to implement. At the BER 10^{-4}, its performance degradation with respect to the MAP decoding algorithm is about $0.6 - 0.7$ dB coding gain loss, against $0.3 - 0.4$ dB loss for the Max-Log-MAP decoding algorithm.

13.3 BIT ERROR PROBABILITY OF MLD

In many practical applications, soft output information is not needed and only the binary decoded codeword is delivered by the decoder. However, it is still desirable to minimize the bit error probability rather than the word error probability. In such cases, the SOVA has no advantage over MLD since both algorithms deliver the same binary decoded sequence. Although the MAP decoding algorithm minimizes the decoding bit error probability, no significant improvement is observed over the bit error probability associated with MLD if properly implemented. Consequently, MLD remains to be the practical solution due to its much lower computational cost and implementation flexibility. However, when a word is in error, different mappings between the information sequences and the code sequences may result in a different number of bits in error, and hence a different average bit error probability.

As described in Chapter 3, the trellis diagram for an (N, K) linear block code is constructed from its TOGM. A trellis-based ML decoder simply finds the most likely path and its corresponding codeword among the 2^K possible paths that represent all the codewords generated by the TOGM. Therefore, a trellis-based decoder can be viewed as a device which searches for the most likely codeword out of the set of the 2^K codewords generated by the TOGM, independent of the mapping between information sequences and codewords. It follows that the mapping between information sequences and the 2^K codewords generated by the TOGM, or equivalently the encoder, can be modified without modifying the trellis-based determination of the most likely codeword. The corresponding information sequence is then retrieved from the knowledge of the mapping used by the encoder. Consequently, a trellis-based ML decoder can be viewed as the cascade of two elements: (1) a trellis search device which delivers the most likely codeword of the code generated by the TOGM; and (2) an inverse mapper which retrieves the information sequence from the delivered codeword. Although all mappings produce the same word error probability, different mappings may produce different bit error probabilities.

Different mappings can be obtained from the TOGM by applying elementary row additions and row permutations to this matrix. Let G_t denote the TOGM of the code considered, and let G_m denote the new matrix. If G_m is used for encoding, then the inverse mapper is represented by the right inverse of G_m. Since this mapping is bijective (or equivalently, G_m has full-rank K) and thus invertible, the right inverse of G_m is guaranteed to exist. In [33], it is shown that for many good codes, the best strategy is to have the K columns of the $K \times K$ identity matrix I_K in G_m (in reduced echelon form). Based on the particular structure of the TOGM G_t, this is readily realized in K steps of Gaussian elimination as follows: For $1 \leq i \leq K$, assume that $G_m(i-1)$ is the matrix obtained at step-$(i-1)$, with $G_m(0) = G_t$ and $G_m(K) = G_m$. Let $c^i = (c_1^i, c_2^i, \ldots, c_K^i)^T$ denote the column of $G_m(i-1)$ that contains the leading '1' of the i-th row of $G_m(i-1)$. Then $c_i^i = 1$ and $c_{i+1}^i = c_{i+2}^i = \cdots = c_K^i = 0$. For $1 \leq j \leq i-1$, add row-i to row-j in $G_m(i-1)$ if $c_j^i = 1$. This results in matrix $G_m(i)$. For $i = K$, $G_m(K) = G_m$ contains the K columns of I_K with the same order of appearance. This matrix is said to be in reduced echelon form (REF), and is referred to as the **REF matrix**.

Now we perform the encoding based on G_m in REF instead of the TOGM G_t. Since both matrices generate the same 2^K codewords, any trellis-based decoder using the trellis diagram constructed from the TOGM G_t can still be used to deliver the most likely codeword. From the knowledge of this codeword and the fact that G_m in REF was used for encoding, the corresponding information sequence is easily recovered by taking only the positions which correspond to the columns of I_K. Note that this strategy is intuitively correct since whenever a code sequence delivered by the decoder is in error, the best strategy to recover the information bits is simply to determine them independently. Otherwise, errors propagate.

Example 13.1 In Example 3.1, the TOGM of the (8,4) RM code is given by

$$G_t = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

After adding rows 2 and 3 to row-1, we obtain

$$G_m = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

We can readily verify that G_t and G_m generate the same 16 codewords, so that there is a one-to-one correspondence between the codewords generated by G_m and the paths in the trellis diagram constructed from G_t. Once the trellis-based decoder delivers the most likely codeword $\hat{u} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_8)$, the corresponding information sequence $\hat{a} = (\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4)$ is retrieved by identifying

$$\begin{aligned} \hat{a}_1 &= \hat{u}_1, \\ \hat{a}_2 &= \hat{u}_2, \\ \hat{a}_3 &= \hat{u}_3, \\ \hat{a}_4 &= \hat{u}_8. \end{aligned}$$

Figure 13.2 depicts the bit error probabilities for the $(32, 26)$ RM code with encoding based on the TOGM and the REF matrix, respectively. The corresponding union bounds obtained in [33] are also shown in this figure. We see that there is a gap in error performance of about 1.0 dB and 0.2 dB at the BERs $10^{-1.5}$ and 10^{-6}, respectively. Similar results have been observed for other good block codes [33, 71]. The situation is different for good convolutional codes of short to medium constraint lengths, for which the feedforward non-systematic realizations outperform their equivalent feedback systematic realizations [80]. This can be explained by the fact that for short to medium constraint length convolutional codes in feedforward form, the bit error probability is dominated by error events of the same structures. Due to the small number of such structures, an efficient mapping that minimizes the bit error probability can be

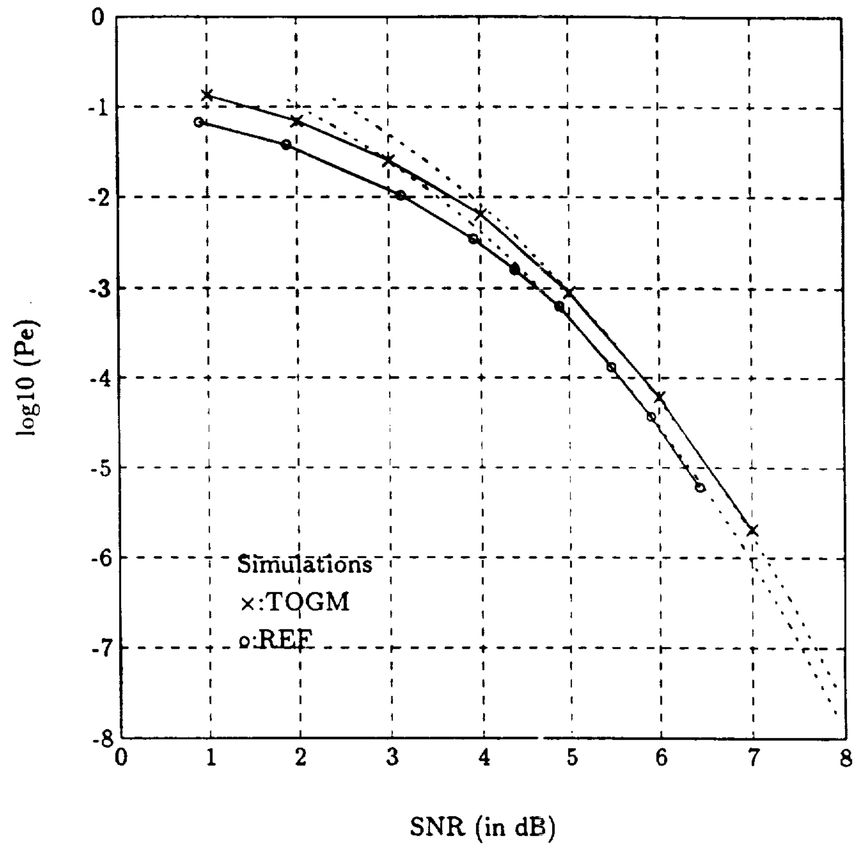

Figure 13.2. The bit error probabilities for the (32, 26) RM code.

devised. This is no longer possible when the error performance is dominated by numerous unstructured error events, such as for long constraint length good convolutional codes or good block codes.

Based on these results, we may conclude: (1) although modest, the difference in bit error performance between encoding with the TOGM and the REF is of the same order as the difference in bit error performances between MLD and some sub-optimum low-complexity decoding methods; (2) the overall error performance of a conventional concatenated scheme with a RS outer code performing algebraic decoding is subject to these differences; and most importantly (3) the gain is free, since only the encoding circuit and the retrieving of the information sequence have to be modified. Furthermore this approach can be used for trellis-based MAP or SOVA decodings if a likelihood measure associated with each bit of the decoded information sequence, rather than each bit of the decoded codeword is needed, as in [34, 39, 40].

This approach can be generalized to any soft decision decoding method. In general, a particular decoding algorithm is based on a particular structure of the code considered. For example, majority-logic-decoding of RM codes is based on the generator matrices of these codes in their original form (presented in Section 2.5), or trellis-based decoding of linear codes is based on the TOGM of the code considered. Two cases are possible depending on whether the decoder delivers a codeword as in trellis-based decoding or directly an information sequence as in majority-logic-decoding of RM codes. In the first case, the procedure previously described is generalized in a straightforward way, while in the second case, the row additions performed to obtain G_m from the generator matrix corresponding to the decoding method considered are applied to the delivered information sequence by the inverse mapper [33].

APPENDIX A
A Trellis Construction Procedure

To decode a linear block code with a trellis-based decoding algorithm, the code trellis must be constructed to be used effectively in the decoding process. Therefore, the construction must meet a number of basic requirements. In the implementation of a trellis-based decoder, every state in the code trellis is labeled. The label of a state is used as the index to the memory where the state metric and the survivor into the state are stored. For efficient indexing, the sequence required to label a state must be as short as possible. Furthermore, the labels of two states at two boundary locations of a trellis section must provide complete information regarding the adjacency of the two states and the label of the composite branch connecting the two states, if they are adjacent, in a simple way. In general, a composite branch label appears many times in a section (see (6.13)). In order to compute the branch metrics efficiently, all the distinct composite branch labels in a trellis section must be generated systematically without duplication and stored in a block of memory. Then, for each pair of adjacent states, the index to the memory storing the composite branch label (or composite branch metric) between the two adjacent states must be derived readily from the labels of the two states. To achieve this, we must derive a condition that two composite branches have the same label, and partition the parallel components in a trellis section into blocks such that the parallel components in a block have the same set of composite branch labels (see Section 6.4). The composite branch label sets for two different blocks are disjoint. This localizes the composite branch metric computation. We can compute the composite branch metrics for each representative parallel component in a block independently. In this appendix, we present an efficient procedure for constructing a sectionalized trellis for a linear block code that meets all the above requirements. The construction makes use of the parallel structure of a trellis section presented in Section 6.4.

A.1 A BRIEF REVIEW OF THE TRELLIS ORIENTED GENERATOR MATRIX FOR A BINARY LINEAR BLOCK CODE

We first give a brief review of the trellis oriented generator matrix (TOGM) for a binary linear block code introduced in Section 3.4. Let G be a binary $K \times N$ generator matrix of a binary (N, K) linear code C. For $1 \le i \le K$, let $\mathrm{ld}(i)$ and $\mathrm{tr}(i)$ denote the column numbers of the leading '1' and the trailing '1' of the i-th row of G, respectively. G is called a TOGM if and only if for $1 \le i < i' \le K$,

$$\mathrm{ld}(i) < \mathrm{ld}(i'), \tag{A.1}$$

$$\mathrm{tr}(i) \neq \mathrm{tr}(i'). \tag{A.2}$$

Let M be a matrix with r rows. Hereafter, for a submatrix M' of M consisting of a subset of the rows in M, the order of rows in M' is assumed to be the same as in M. For a submatrix M' of M consisting of the i_1-th row, the i_2-th row, ..., the i_p-th row of M, let us call the set $\{i_1, i_2, \ldots, i_p\}$ as the **row number set** of M' (as a submatrix of M). M is said to be partitioned into the submatrices $M_1, M_2, \ldots, M_\mu$ if each row of M is contained in exactly one submatrix M_i with $1 \le i \le \mu$.

In Section 3.4, a TOGM G of C is partitioned into three submatrices, G_h^p, G_h^f, and G_h^s (also shown in Figure 6.3), for $0 \le h \le N$. The row number sets of G_h^p, G_h^f, and G_h^s are $\{i : \mathrm{tr}(i) \le h\}$, $\{i : h < \mathrm{ld}(i)\}$, $\{i : \mathrm{ld}(i) \le h < \mathrm{tr}(i)\}$, respectively. G_h^p and G_h^f generate the past and future codes at time-h, $C_{0,h}$ and $C_{h,N}$, respectively (see Section 3.7). That is,

$$C_{0,h} = \Gamma(G_h^p), \tag{A.3}$$

$$C_{h,N} = \Gamma(G_h^f), \tag{A.4}$$

where for a matrix M, $\Gamma(M)$ denotes the linear space generated by M.

Since from (A.1) $\mathrm{ld}(i) < \mathrm{ld}(i')$ for $1 \le i < i' \le K$, the order of information bits corresponds to the order of rows in G, and therefore, the row number sets of G_h^p, G_h^s and G_h^f as submatrices of G correspond to A_h^p, A_h^s and A_h^f (refer to Section 3.4), respectively. In this appendix, we put the TOGM in reverse direction such that for $1 \le i < i' \le K$,

$$\mathrm{ld}(i) \neq \mathrm{ld}(i'), \tag{A.5}$$

$$\mathrm{tr}(i) > \mathrm{tr}(i'). \tag{A.6}$$

Using a TOGM in reverse order, we can store the states (or state metrics) at the left end of a parallel component in consecutive memory cells using a simple addressing method. This method is very useful for designing IC decoder. It also reduces the actual computation time for a software decoder, since (the metrics of) the states at the left ends are accessed consecutively and computers have cache memories.

For a binary m-tuple $u = (u_1, u_2, \ldots, u_m)$ and a set $I = \{i_1, i_2, \ldots, i_p\}$ of p positive integers with $1 \le i_1 < i_2 < \cdots < i_p \le m$, define

$$p_I(u) \triangleq (u_{i_1}, u_{i_2}, \ldots, u_{i_p}). \tag{A.7}$$

For convenience, for $I = \emptyset$, $p_I(u) \triangleq \epsilon$ (the null sequence). For a set U of m-tuples, define

$$p_I(U) \triangleq \{p_I(u) : u \in U\}. \tag{A.8}$$

For $I = \{h+1, h+2, \ldots, h'\}$, p_I is denoted by $p_{h,h'}$.

Then, for a partition $\{M_1, M_2, \ldots, M_\mu\}$ of M with r rows and $v \in \{0,1\}^r$, it follows from the definition of a row number set and (A.7) that

$$vM = p_{I_1}(v)M_1 + p_{I_2}(v)M_2 + \cdots + p_{I_\mu}(v)M_\mu, \tag{A.9}$$

where I_i denotes the row number set of M_i for $1 \le i \le \mu$. If M_i is further partitioned into submatrices $M_{i_1}, M_{i_2}, \ldots$, then

$$p_{I_i}(v)M_i = p_{I_{i_1}}(p_{I_i}(v))M_{i_1} + p_{I_{i_2}}(p_{I_i}(v))M_{i_2} + \cdots, \tag{A.10}$$

where $I_{i_1}, I_{i_2}, \ldots$, denote the row number sets of $M_{i_1}, M_{i_2}, \ldots$ as submatrices of M_i.

In the construction of the trellis section from time-h to time-h', the submatrix G_h^s of G takes a key role. Let I_h denote the row number set of G_h^s as a submatrix G. Then, $\rho_h = |I_h|$, and the following property of state labeling by the state defining information set (Section 4.1) holds.

The Key Property of State Labeling: Let $L(\sigma_0, \sigma_h, \sigma_f) \triangleq L(\sigma_0, \sigma_h) \circ L(\sigma_h, \sigma_f)$ be the set of paths in a code trellis for C that connects the initial state σ_0 to the final state σ_f through the state σ_h at time-h. There is a one-to-one mapping l from $\Sigma_h(C)$ to $\{0,1\}^{\rho_h}$ with $0 < h < N$ such that for $\sigma_h \in \Sigma_h(C)$ and $a \in \{0,1\}^K$,

$$aG \in L(\sigma_0, \sigma_h, \sigma_f), \tag{A.11}$$

if and only if

$$p_{I_h}(a) = l(\sigma_h). \tag{A.12}$$

$\triangle\triangle$

Here, $l(\sigma_h) \in \{0,1\}^{\rho_h}$ is called the label of state σ_h. This state labeling is simply the state labeling by the state defining information set given in Section 4.1. We can readily see that

$$L(\sigma_0, \sigma_h, \sigma_f) = l(\sigma_h) G_h^s \oplus C_{0,h} \oplus C_{h,N}. \tag{A.13}$$

Note that if $\rho_h = 0$, G_h^s is the empty matrix and $l(\sigma_h) = \epsilon$. For convenience, we define the product of ϵ and the empty matrix as the zero vector.

A.2 STATE LABELING BY THE STATE DEFINING INFORMATION SET AND COMPOSITE BRANCH LABEL

For $\sigma_h \in \Sigma_h(C)$ and $\sigma_{h'} \in \Sigma_{h'}(C)$ with $L(\sigma_h, \sigma_{h'}) \neq \emptyset$, the composite branch label $L(\sigma_h, \sigma_{h'})$ can be expressed in terms of the labels of σ_h and $\sigma_{h'}$. Since

$$\begin{aligned} L(\sigma_0, \sigma_h, \sigma_{h'}, \sigma_f) &\triangleq L(\sigma_0, \sigma_h) \circ L(\sigma_h, \sigma_{h'}) \circ L(\sigma_{h'}, \sigma_f) \\ &= L(\sigma_0, \sigma_h, \sigma_f) \cap L(\sigma_0, \sigma_{h'}, \sigma_f), \end{aligned}$$

it follows from the key property of state labeling that for $a \in \{0,1\}^K$,

$$aG \in L(\sigma_0, \sigma_h, \sigma_{h'}, \sigma_f), \tag{A.14}$$

if and only if

$$p_{I_h}(a) = l(\sigma_h), \tag{A.15}$$

$$p_{I_{h'}}(a) = l(\sigma_{h'}). \tag{A.16}$$

Define $I_{h,h'}$ and $\rho_{h,h'}$ as follows:

$$I_{h,h'} = I_h \cap I_{h'} \tag{A.17}$$

$$\rho_{h,h'} = |I_{h,h'}|. \tag{A.18}$$

Then $I_{h,h'} = \{i : \mathrm{ld}(i) \leq h < h' < \mathrm{tr}(i)\}$ is the row number set of the submatrix of G which consists of those rows in both G_h^s and $G_{h'}^s$. Let $G_{h,h'}^{s,s}$ denote this submatrix. Let $G_{h,h'}^{s,p}$, $G_{h,h'}^{f,s}$ and $G_{h,h'}^{f,p}$ denote the submatrices of G whose row number sets as submatrices of G are $I_h \backslash I_{h,h'} = \{i : \mathrm{ld}(i) \leq h < \mathrm{tr}(i) \leq h'\}$, $I_{h'} \backslash I_{h,h'} = \{i : h < \mathrm{ld}(i) \leq h' < \mathrm{tr}(i)\}$ and $\{i : h < \mathrm{ld}(i) \leq \mathrm{tr}(i) \leq h'\}$, respectively. Then,

$$C_{h,h'} = \Gamma(G_{h,h'}^{f,p}), \tag{A.19}$$

and G is partitioned into G_h^p, $G_{h,h'}^{s,s}$, $G_{h,h'}^{s,p}$, $G_{h,h'}^{f,s}$, $G_{h,h'}^{f,p}$ and $G_{h'}^f$ (see Figure 6.3). From (A.3), (A.4), (A.9) and (A.19), we have that for $a \in \{0,1\}^K$, $aG \in C$ can be expressed as

$$aG = p_{I_{h,h'}}(a)G_{h,h'}^{s,s} + p_{I_h \backslash I_{h,h'}}(a)G_{h,h'}^{s,p} + p_{I_{h'} \backslash I_{h,h'}}(a)G_{h,h'}^{f,s} + u, \tag{A.20}$$

where $u \in C_{0,h} \oplus C_{h,h'} \oplus C_{h',N}$.

Since h and h' are fixed hereafter, we abbreviate $p_{h,h'}(G_{h,h'}^{x,y})$ as $G^{x,y}$ where $x \in \{s,f\}$ and $y \in \{s,p\}$. If $aG \in L(\sigma_0, \sigma_h, \sigma_{h'}, \sigma_f)$, then $p_{h,h'}(aG) = ap_{h,h'}(G) \in L(\sigma_h, \sigma_{h'})$. Since $L(\sigma_h, \sigma_{h'})$ is a coset in $p_{h,h'}(C)/C_{h,h'}^{\mathrm{tr}}$ (see (3.18)), it follows from (A.20) that

$$L(\sigma_h, \sigma_{h'}) = p_{I_{h,h'}}(a)G^{s,s} + p_{I_h \backslash I_{h,h'}}(a)G^{s,p} + p_{I_{h'} \backslash I_{h,h'}}(a)G^{f,s} + C_{h,h'}^{\mathrm{tr}}. \tag{A.21}$$

In the following, we will derive a relation between $L(\sigma_h, \sigma_{h'})$ and the labels of states σ_h and $\sigma_{h'}$, $l(\sigma_h)$ and $l(\sigma_{h'})$. Lemma A.1 gives a simple relation between $I_{h,h'}$ and I_h.

Lemma A.1 $I_{h,h'}$ consists of the smallest $\rho_{h,h}$ integers in I_h.

Proof: For $i \in I_{h,h'}$, we have that $h' < \mathrm{tr}(i)$. For $i' \in I_h \backslash I_{h,h'}$, we have that $\mathrm{tr}(i') \leq h'$. Hence, $\mathrm{tr}(i') < \mathrm{tr}(i)$. From (A.6), we have $i < i'$.

$\triangle\triangle$

In general, there is no such a simple relation between $I_{h,h'}$ and $I_{h'}$.

Let $I_{h'} = \{i_1, i_2, \ldots, i_{\rho_{h'}}\}$ with $1 \leq i_1 < i_2 < \cdots < i_{\rho_{h'}}$ be the row number set of $G^s_{h'}$, and let $I_{h,h'} = \{i_{j_1}, i_{j_2}, \ldots, i_{j_{\rho_{h,h'}}}\}$ with $1 \leq j_1 < j_2 \cdots < j_{\rho_{h,h'}} \leq \rho_{h'}$ be the row number set of $G^{s,s}_{h,h'}$. By definition, the ρ-th row of $G^s_{h'}$ is the i_ρ-th row of G for $1 \leq \rho \leq \rho_{h'}$ and the μ-th row of $G^{s,s}_{h,h'}$ is the i_{j_μ}-th row of G for $1 \leq \mu \leq \rho_{h,h'}$. Hence, the μ-th row of $G^{s,s}_{h,h'}$ is the j_μ-th row of $G^s_{h'}$. That is, the row number set of $G^{s,s}_{h,h'}$ as a submatrix of $G^s_{h'}$, denoted $J_{h'}$, is $J_{h'} = \{j_1, j_2, \ldots, j_{\rho_{h,h'}}\}$, and $\bar{J}_{h'} \triangleq \{1, 2, \ldots, \rho_{h'}\} \setminus J_{h'}$ is the row number set of $G^{f,s}_{h,h'}$.

Suppose (A.14) holds. Then, from (A.15), (A.16) and Lemma A.1,

$$p_{I_{h,h'}}(a) = p_{0,\rho_{h,h'}}(l(\sigma_h)) = p_{J_{h'}}(l(\sigma_{h'})), \tag{A.22}$$

$$p_{I_h \backslash I_{h,h'}}(a) = p_{\rho_{h,h'},\rho_h}(l(\sigma_h)), \tag{A.23}$$

$$p_{I_{h'} \backslash I_{h,h'}}(a) = p_{\bar{J}_{h'}}(l(\sigma_{h'})). \tag{A.24}$$

For simplicity, define

$$l^{(s)}(\sigma_h) \triangleq p_{0,\rho_{h,h'}}(l(\sigma_h)), \tag{A.25}$$

$$l^{(p)}(\sigma_h) \triangleq p_{\rho_{h,h'},\rho_h}(l(\sigma_h)), \tag{A.26}$$

$$l^{(s)}(\sigma_{h'}) \triangleq p_{J_{h'}}(l(\sigma_{h'})), \tag{A.27}$$

$$l^{(f)}(\sigma_{h'}) \triangleq p_{\bar{J}_{h'}}(l(\sigma_{h'})). \tag{A.28}$$

From (A.25) and (A.26), the label $l(\sigma_h)$ of the state σ_h with $l^{(s)}(\sigma_h) = \alpha$ and $l^{(p)}(\sigma_h) = \beta$ is given by

$$l(\sigma_h) = \alpha \circ \beta. \tag{A.29}$$

The label $l(\sigma_{h'})$ of the state $\sigma_{h'}$ with $l^{(s)}(\sigma_h) = \alpha$ and $l^{(f)}(\sigma_{h'}) = \gamma$ can be easily obtained from α and γ using (A.27) and (A.28).

By summarizing (A.14) to (A.16), (A.22) to (A.26) and (A.28), a condition for the adjacency between two states at time-h and time-h', and the composite branch label between them (shown in Section 6.3) are given in Theorem A.1.

Theorem A.1 For $\sigma_h \in \Sigma_h(C)$ and $\sigma_{h'} \in \Sigma_{h'}(C)$ with $0 \leq h < h' \leq N$, $L(\sigma_h, \sigma_{h'}) \neq \emptyset$ if and only if

$$l^{(s)}(\sigma_h) = l^{(s)}(\sigma_{h'}).$$

Also if $L(\sigma_h, \sigma_{h'}) \neq \emptyset$, then $L(\sigma_h, \sigma_{h'})$ is given by

$$L(\sigma_h, \sigma_{h'}) = l^{(s)}(\sigma_h)G^{s,s} + l^{(p)}(\sigma_h)G^{s,p} + l^{(f)}(\sigma_{h'})G^{f,s} + C^{\mathrm{tr}}_{h,h'}. \tag{A.30}$$

$\triangle\triangle$

We call $l^{(s)}(\sigma_h)$ (or $l^{(s)}(\sigma_{h'})$) the first label part of σ_h (or $\sigma_{h'}$) and $l^{(p)}(\sigma_h)$ (or $l^{(f)}(\sigma_{h'})$) the second label part of σ_h (or $\sigma_{h'}$). Now we partition $\Sigma_h(C)$ and $\Sigma_{h'}(C)$ into $2^{\rho_{h,h'}}$ blocks of the same size, respectively, in such a way that two states are in the same block if and only if they have the same first label part. For $\alpha \in \{0,1\}^{\rho_{h,h'}}$, a block, denoted Σ_h^{α}, (or $\Sigma_{h'}^{\alpha}$) in the above partition of $\Sigma_h(C)$ (or $\Sigma_{h'}(C)$) is defined as

$$\Sigma_h^{\alpha} \triangleq \{\sigma_h \in \Sigma_h(C) : l^{(s)}(\sigma_h) = \alpha\}, \tag{A.31}$$

$$\Sigma_{h'}^{\alpha} \triangleq \{\sigma_{h'} \in \Sigma_{h'}(C) : l^{(s)}(\sigma_{h'}) = \alpha\}. \tag{A.32}$$

The blocks Σ_h^{α} and $\Sigma_{h'}^{\alpha}$ correspond to $S_L(a_h)$ and $S_R(a_h)$ in Section 6.4, respectively. That is, for any $\alpha \in \{0,1\}^{\rho_{h,h'}}$, a subgraph which consists of the set of state at time-h, Σ_h^{α}, the set of states at time-h', $\Sigma_{h'}^{\alpha}$, and all the composite branches between them form a parallel component. This parallel component is denoted Λ_{α}. The total number of parallel components in a trellis section from time-h to time-h' is given by $2^{\rho_{h,h'}}$.

Since all states in $\Sigma_h^{\alpha} \cup \Sigma_{h'}^{\alpha}$ have the same first label part α, there is a one-to-one mapping, denoted $s_{h,\alpha}$, from the set of the second label parts of the states in Σ_h^{α}, denoted L_h, to Σ_h^{α}, and there is a one-to-one mapping, denoted $s_{h',\alpha}$, from the set of the second label parts of the states in $\Sigma_{h'}^{\alpha}$, denoted $L_{h'}$, to $\Sigma_{h'}^{\alpha}$. Then, $L_h = \{0,1\}^{\rho_h - \rho_{h,h'}}$, $L_{h'} = \{0,1\}^{\rho_{h'} - \rho_{h,h'}}$ and $s_{h,\alpha}$ and $s_{h',\alpha}$ are the inverse mappings of $l^{(p)}$ and $l^{(f)}$, respectively.

A.3 TRELLIS CONSTRUCTION

Now consider how to construct a trellis section. When we use a trellis diagram for decoding, we have to store the state metric for each state. Therefore, we must assign memory for each state. For a state σ, we use the label $l(\sigma)$ as the index of the allocated memory, because: (1) $l(\sigma)$ is the shortest possible label for σ and is easy to obtain, and (2) as shown in Theorem A.1, the complete information on $L(\sigma_h, \sigma_{h'})$ is provided in a very simple way by $l^{(s)}(\sigma_h)$, $l^{(p)}(\sigma_h)$, $l^{(s)}(\sigma_{h'})$ and $l^{(f)}(\sigma_{h'})$ which can be readily derived from $l(\sigma_h)$ and $l(\sigma_{h'})$ by (A.25) to (A.28).

If instead we use the state labeling by parity-check matrix for the states at time-h, the label length becomes $N-K$. Since $\rho_h \leq \min\{N, N-K\}$ (see (5.4)), if $\rho_h < N-K$, a linear mapping from $\{0,1\}^{N-K}$ to $\{0,1\}^{\rho_h}$ which depends on h in general is necessary to obtain the index of the memory.

The next problem is how to generate branch metrics of adjacent state pairs and give access to them. The set of composite branch labels of the section from time-h to time-h', denoted $L_{h,h'}$, is the set of the following cosets (see (3.18)):

$$L_{h,h'} = p_{h,h'}(C)/C^{\text{tr}}_{h,h'}. \tag{A.33}$$

It follows from (A.30) that

$$\begin{aligned} L_{h,h'} \;=\; & \{\alpha G^{s,s} + \beta G^{s,p} + \gamma G^{f,s} + C^{\text{tr}}_{h,h'} : \alpha \in \{0,1\}^{\rho_{h,h'}}, \\ & \beta \in \{0,1\}^{\rho_h - \rho_{h,h'}}, \gamma \in \{0,1\}^{\rho_{h'} - \rho_{h,h'}}\}. \end{aligned} \tag{A.34}$$

Each composite branch label appears

$$2^{K - k(C_{0,h}) - k(C_{h',N}) - k(p_{h,h'}(C))}$$

times in the trellis section (see (6.13)).

Next we consider how to generate all the composite branch labels in $L_{h,h'}$ given by (A.34) without duplication. Suppose we choose submatrices $G_1^{s,s}$ of $G^{s,s}$, $G_1^{s,p}$ of $G^{s,p}$ and $G_1^{f,s}$ of $G^{f,s}$ such that the rows in $G_1^{s,s}$, $G_1^{s,p}$, $G_1^{f,s}$ and $G^{f,p}$ are linearly independent and

$$\begin{aligned} L_{h,h'} \;=\; & \{\alpha_1 G^{s,s} + \beta_1 G^{s,p} + \gamma_1 G^{f,s} + C^{\text{tr}}_{h,h'} : \alpha_1 \in \{0,1\}^{r_{s,s}}, \\ & \beta_1 \in \{0,1\}^{r_{s,p}}, \gamma_1 \in \{0,1\}^{r_{f,s}}\}, \end{aligned} \tag{A.35}$$

where $r_{s,s}$, $r_{s,p}$ and $r_{f,s}$ denote the numbers of rows of $G_1^{s,s}$, $G_1^{s,p}$ and $G_1^{f,s}$, respectively. If we generate composite branch labels by using the right-hand side of (A.35) and store $\alpha_1 G^{s,s} + \beta_1 G^{s,p} + \gamma_1 G^{f,s} + C^{\text{tr}}_{h,h'}$ into a memory indexed

with $(\alpha_1, \beta_1, \gamma_1)$, then the duplication can be avoided. The next requirement is that the index $(\alpha_1, \beta_1, \gamma_1)$ can be readily obtained from α, β and γ. To provide a good solution of the above problem, we first analyze the linear dependency of rows in $p_{h,h'}(G)$.

For a TOGM G, $p_{h,h'}(G)$ consists of the disjoint submatrices $G^{s,s}$, $G^{s,p}$, $G^{f,s}$, $G^{f,p}$ and the all zero submatrices $p_{h,h'}(G^{p}_{h})$ and $p_{h,h'}(G^{f}_{h'})$.

Lemma A.2 In $p_{h,h'}(G)$: (1) the rows in submatrices $G^{f,s}$ and $G^{f,p}$ are linearly independent; and (2) the rows in submatrices $G^{s,p}$ and $G^{f,p}$ are linearly independent.

Proof: If the i-th row of G is in $G^{f,s}$ or $G^{f,p}$, then

$$h < \mathrm{ld}(i) \leq h'. \tag{A.36}$$

Similarly, if the i-th row of G is in $G^{s,p}$ or $G^{f,p}$, then

$$h < \mathrm{tr}(i) \leq h'. \tag{A.37}$$

Hence, (1) and (2) of the lemma follow from (A.1) and (A.2), respectively.

△△

For two binary $r \times m$ matrices M and M' and a binary linear block code C_0 of length m, we write

$$M \equiv M' \qquad (\mathrm{mod} C_0)$$

if and only if every row in the matrix $M - M'$ is in C_0, where "$-$" denotes the component-wise subtraction.

Partition $G^{f,s}$ into two submatrices $G^{f,s}_0$ and $G^{f,s}_1$ (see Figure A.1) by partitioning the rows of $G^{f,s}$ in such a way that

(1) the rows in $G^{f,s}_1$, $G^{s,p}$ and $G^{f,p}$ are linearly independent, and

(2) each row of $G^{f,s}_0$ is a linear combination of rows in $G^{f,s}_1$, $G^{s,p}$ and $G^{f,p}$.

Let ν_1 denote the number of rows of $G^{f,s}_1$, and define $\nu_0 \triangleq \rho_{h'} - \rho_{h,h'} - \nu_1$, which is the number of rows of $G^{f,s}_0$.

Let $e_1, e_1, \ldots, e_{\nu_0}$ be the first to the last rows of $G^{f,s}_0$. From conditions (1) and (2) of the partitioning of $G^{f,s}$, there are unique $v^{(1)}_i \in \{0,1\}^{\nu_1}$, $v^{(2)}_i \in \{0,1\}^{\rho_h - \rho_{h,h'}}$ and $u_i \in C^{tr}_{h,h'}$ such that

$$e_i = v^{(1)}_i G^{f,s}_1 + v^{(2)}_i G^{s,p} + u_i, \quad \text{for } 1 \leq i \leq \nu_0.$$

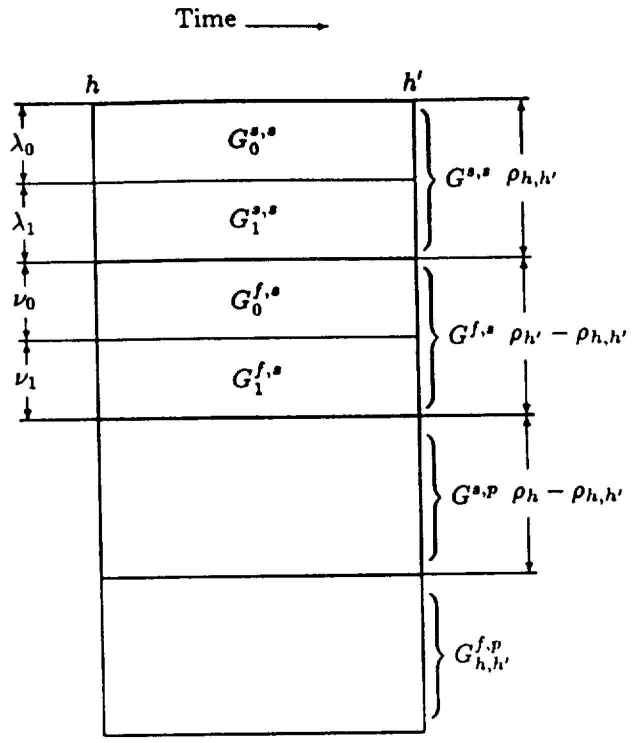

Figure A.1. Further partitioning of the TOGM G.

Let $E^{(1)}$ denote the $\nu_0 \times \nu_1$ matrix whose i-th row is $v_i^{(1)}$ and $E^{(2)}$ denote the $\nu_0 \times (\rho_h - \rho_{h,h'})$ matrix whose i-th row is $v_i^{(2)}$. Then, we have

$$G_0^{f,s} \equiv E^{(1)}G_1^{f,s} + E^{(2)}G^{s,p} \quad (\text{mod} C_{h,h'}^{tr}). \tag{A.38}$$

$G_0^{f,s}$, $G_1^{f,s}$, $E^{(1)}$ and $E^{(2)}$ can be efficiently derived by using standard row operations.

Next, partition $G^{s,s}$ into two submatrices $G_0^{s,s}$ and $G_1^{s,s}$ (see Figure A.1) by partitioning the rows of $G^{s,s}$ in such a way that:

(1) the rows in $G_1^{s,s}$, $G_1^{f,s}$, $G^{s,p}$ and $G^{f,p}$ are linearly independent; and

(2) each row of $G_0^{s,s}$ is a linear combination of rows in $G_1^{s,s}$, $G_1^{f,s}$, $G^{s,p}$ and $G^{f,p}$.

Let λ_1 denote the number of rows of $G_1^{s,s}$, and define $\lambda_0 \triangleq \rho_{h,h'} - \lambda_1$, which is the number of rows of $G_0^{s,s}$. In a similar way as the derivation of (A.38), we can find a unique $\lambda_0 \times \lambda_1$ matrix $F^{(1)}$, a unique $\lambda_0 \times \nu_1$ matrix $F^{(2)}$ and a unique $\lambda_0 \times (\rho_h - \rho_{h,h'})$ matrix $F^{(3)}$ such that

$$G_0^{s,s} \equiv F^{(1)}G_1^{s,s} + F^{(2)}G_1^{f,s} + F^{(3)}G^{s,p} \quad (\text{mod} C_{h,h'}^{\text{tr}}). \qquad \text{(A.39)}$$

Let R and Q denote the row number sets of $G_1^{f,s}$ and $G_1^{s,s}$, respectively. Define $\bar{R} \triangleq \{1, 2, \ldots, \rho_{h'} - \rho_{h,h'}\} \backslash R$ and $\bar{Q} \triangleq \{1, 2, \ldots, \rho_{h,h'}\} \backslash Q$. Then, from (A.9)

$$\begin{aligned} \alpha G^{s,s} &= p_{\bar{Q}}(\alpha)G_0^{s,s} + p_Q(\alpha)G_1^{s,s}, \quad \text{for } \alpha \in \{0,1\}^{\rho_{h,h'}}, & \text{(A.40)} \\ \gamma G^{f,s} &= p_{\bar{R}}(\gamma)G_0^{f,s} + p_R(\gamma)G_1^{f,s}, \quad \text{for } \gamma \in \{0,1\}^{\rho_{h'} - \rho_{h,h'}}. & \text{(A.41)} \end{aligned}$$

It follows from (A.38) to (A.41) that for $\alpha \in \{0,1\}^{\rho_{h,h'}}$, $\beta \in \{0,1\}^{\rho_h - \rho_{h,h'}}$ and $\gamma \in \{0,1\}^{\rho_{h'} - \rho_{h,h'}}$,

$$\begin{aligned} &\alpha G^{s,s} + \beta G^{s,p} + \gamma G^{f,s} + C_{h,h'}^{\text{tr}} \\ &= (p_Q(\alpha) + p_{\bar{Q}}(\alpha)F^{(1)})G_1^{s,s} \\ &\quad + (p_{\bar{Q}}(\alpha)F^{(3)} + p_{\bar{R}}(\gamma)E^{(2)} + \beta)G^{s,p} \\ &\quad + (p_{\bar{Q}}(\alpha)F^{(2)} + p_R(\gamma) + p_{\bar{R}}(\gamma)E^{(1)})G_1^{f,s} + C_{h,h'}^{\text{tr}}. \qquad \text{(A.42)} \end{aligned}$$

Define

$$\begin{aligned} f^{(1)}(\alpha) &\triangleq p_Q(\alpha) + p_{\bar{Q}}(\alpha)F^{(1)}, & \text{(A.43)} \\ f^{(2)}(\alpha, \beta, \gamma) &\triangleq p_{\bar{Q}}(\alpha)F^{(3)} + p_{\bar{R}}(\gamma)E^{(2)} + \beta, & \text{(A.44)} \\ f^{(3)}(\alpha, \gamma) &\triangleq p_{\bar{Q}}(\alpha)F^{(2)} + p_R(\gamma) + p_{\bar{R}}(\gamma)E^{(1)}. & \text{(A.45)} \end{aligned}$$

When α, β and γ run over $\{0,1\}^{\rho_{h,h'}}$, $\{0,1\}^{\rho_h - \rho_{h,h'}}$ and $\{0,1\}^{\rho_{h'} - \rho_{h,h'}}$, respectively, $f^{(1)}(\alpha)$ $f^{(2)}(\alpha, \beta, \gamma)$ and $f^{(3)}(\alpha, \gamma)$ run over $\{0,1\}^{\lambda_1}$, $\{0,1\}^{\rho_h - \rho_{h,h'}}$ and $\{0,1\}^{\nu_1}$, respectively. Hence, the set $L_{h,h'}$ of all composite branch labels is given by

$$\begin{aligned} L_{h,h'} = \{&\alpha_1 G_1^{s,s} + \beta_1 G^{s,p} + \gamma_1 G_1^{f,s} + C_{h,h'}^{\text{tr}} : \\ &\alpha_1 \in \{0,1\}^{\lambda_1}, \beta_1 \in \{0,1\}^{\rho_h - \rho_{h,h'}}, \gamma_1 \in \{0,1\}^{\nu_1}\}. \qquad \text{(A.46)} \end{aligned}$$

Since the rows in $G_1^{s,s}$, $G^{s,p}$, $G_1^{f,s}$ and $G^{f,p}$ are linearly independent, for $(\alpha_1, \beta_1, \gamma_1) \neq (\alpha_1', \beta_1', \gamma_1')$, the cosets $\alpha_1 G_1^{s,s} + \beta_1 G^{s,p} + \gamma_1 G_1^{f,s} + C_{h,h'}^{\text{tr}}$ and

$\alpha_1' G_1^{s,s} + \beta_1' G^{s,p} + \gamma_1' G_1^{f,s} + C_{h,h'}^{\text{tr}}$ are disjoint. Consequently, the set of composite branch labels of a parallel component Λ_{α} with $\alpha \in \{0,1\}^{\rho_{h,h'}}$, denoted L_{α}, is given by

$$\begin{aligned} L_{\alpha} \;\triangleq\; & \{\alpha_1 G_1^{s,s} + \beta_1 G^{s,p} + \gamma_1 G_1^{f,s} + C_{h,h'}^{\text{tr}} : \\ & \beta_1 \in \{0,1\}^{\rho_h - \rho_{h,h'}}, \gamma_1 \in \{0,1\}^{\nu_1}\}, \end{aligned} \tag{A.47}$$

where $\alpha_1 = f^{(1)}(\alpha) = p_Q(\alpha) + p_{\bar{Q}}(\alpha) F^{(1)}$.

Theorem A.2 Let Q denote the row number sets of $G_1^{s,s}$, and define $\bar{Q} \triangleq \{1,2,\ldots,\rho_{h,h'}\} \setminus Q$. Two parallel components Λ_{α} and $\Lambda_{\alpha'}$ with α and α' in $\{0,1\}^{\rho_{h,h'}}$ are isomorphic up to composite branch labels, if and only if

$$p_Q(\alpha + \alpha') = (p_{\bar{Q}}(\alpha + \alpha')) F^{(1)}, \tag{A.48}$$

where $F^{(1)}$ is defined in (A.39). If (A.48) does not hold, L_{α} and $L_{\alpha'}$ are disjoint.

Proof: (1) The only-if part: If $\alpha_1 \triangleq p_Q(\alpha) + p_{\bar{Q}}(\alpha) F^{(1)} \neq \alpha_1' \triangleq p_Q(\alpha') + p_{\bar{Q}}(\alpha') F^{(1)}$, then L_{α} and $L_{\alpha'}$ are mutually disjoint from (A.47).

(2) The if part: Suppose that (A.48) holds, that is, $\alpha_1 = \alpha_1'$. Let $l_{\alpha+\alpha'}$ denote the binary $(\rho_{h'} - \rho_{h,h'})$-tuple such that

$$p_R(l_{\alpha+\alpha'}) = p_{\bar{Q}}(\alpha + \alpha') F^{(2)}, \tag{A.49}$$

$$p_{\bar{R}}(l_{\alpha+\alpha'}) = 0. \tag{A.50}$$

For any given state pair $(\sigma_h, \sigma_{h'}) \in \Sigma_h^{\alpha} \times \Sigma_{h'}^{\alpha}$, define $(\sigma_h', \sigma_{h'}') \in \Sigma_h^{\alpha'} \times \Sigma_{h'}^{\alpha'}$ as

$$\begin{aligned} l^{(p)}(\sigma_h') &= l^{(p)}(\sigma_h) + p_{\bar{Q}}(\alpha + \alpha') F^{(3)}, \\ & \qquad \text{i.e.,} \sigma_h' = s_{h,\alpha'}(l^{(p)}(\sigma_h) + p_{\bar{Q}}(\alpha + \alpha') F^{(3)}), \quad \text{(A.51)} \\ l^{(f)}(\sigma_{h'}') &= l^{(f)}(\sigma_{h'}) + l_{\alpha+\alpha'}, \\ & \qquad \text{i.e,} \sigma_{h'}' = s_{h',\alpha'}(l^{(f)}(\sigma_{h'}) + l_{\alpha+\alpha'}). \quad \text{(A.52)} \end{aligned}$$

Then, $f^{(1)}(\alpha) = f^{(1)}(\alpha')$, $f^{(2)}(\alpha, l^{(p)}(\sigma_h), l^{(f)}(\sigma_{h'})) = f^{(2)}(\alpha', l^{(p)}(\sigma_h'), l^{(f)}(\sigma_{h'}'))$ and $f^{(3)}(\alpha, l^{(f)}(\sigma_{h'})) = f^{(3)}(\alpha', l^{(f)}(\sigma_{h'}'))$. Hence, it follows from (A.30), and (A.42) to (A.45) that

$$L(\sigma_h, \sigma_{h'}) = L(\sigma_h', \sigma_{h'}').$$

Note that when σ_h runs over Σ_h^{α}, $l^{(p)}(\sigma_h)$ runs over $L_h = \{0,1\}^{\rho_h - \rho_{h,h'}}$ and therefore, $s_{h,\alpha'}(l^{(p)}(\sigma_h) + p_{\bar{Q}}(\alpha + \alpha')F^{(3)})$ defines a permutation of $\Sigma_h^{\alpha'}$. Similarly, $s_{h',\alpha'}(l^{(f)}(\sigma_{h'}) + l_{\alpha+\alpha'})$ defines a permutation of $\Sigma_{h'}^{\alpha'}$.

△△

Corollary A.1 The block of the isomorphic parallel components containing Λ_α is given by

$$\{\Lambda_{\alpha+\alpha'} : p_Q(\alpha') = p_{\bar{Q}}(\alpha')F^{(1)}, p_{\bar{Q}}(\alpha') \in \{0,1\}^{\lambda_0}\}. \tag{A.53}$$

Each block of the partition consists of 2^{λ_0} identical parallel components, where λ_0 is the number of rows of $G_0^{s,s}$.

△△

It is shown in [44] that λ_0 is equal to $\lambda_{h,h'}(C)$ defined by (6.36).

Example A.1 Consider the $\mathrm{RM}_{3,6}$ code which is a (64,42) code. The second section of the 8-section minimal trellis diagram $T(\{0,8,16,\ldots,64\})$ for this code consists of 16 parallel components, and they are partitioned into two blocks. Each block of the partition consists of 2^3 identical parallel components, Each parallel components has 8 states at time 8 and 64 states at time 16. Hence, there are $2^{13} = 16 \times 8 \times 64$ composite branches in this trellis section. However, there are only $2^{10} = 2 \times 8 \times 64$ different composite branch labels.

△△

A.4 AN EFFICIENT TRELLIS CONSTRUCTION PROCEDURE

In this section, an efficient procedure for constructing the trellis section from time-h to time-h' is presented. First, we present a subprocedure, denoted **GenerateCBL**(α), that generates the set of composite branch labels for a representative parallel component Λ_α in a block. From Corollary A.1, we can choose the parallel component Λ_α with $p_{\bar{Q}}(\alpha) = \mathbf{0}$ as the representative (in (A.53), for any α, $\Lambda_{\alpha+\alpha'}$ with $p_{\bar{Q}}(\alpha') = p_{\bar{Q}}(\alpha)$ is such one). Let $\alpha_1 \triangleq p_Q(\alpha)$. Then, the subprocedure, GenerateCBL(α_1), generates the set of the composite branch labels of the parallel component Λ_α, denoted L_{α_1}:

$$\begin{aligned} L_{\alpha_1} &= \{\alpha_1 G_1^{s,s} + \gamma_1 G_1^{f,s} + \beta G^{s,p} + C_{h,h'}^{tr} : \\ &\qquad \gamma_1 \in \{0,1\}^{\nu_1}, \beta \in \{0,1\}^{\rho_h - \rho_{h,h'}}\}, \end{aligned}$$

and stores each composite branch label $\alpha_1 G_1^{s,s} + \gamma_1 G_1^{f,s} + \beta G^{s,p} + C_{h,h'}^{tr}$, in the memory at the index $\alpha_1 \circ \gamma_1 \circ \beta$. This makes it possible to store the composite branch labels in the parallel component at consecutive memory locations.

It follows from (A.42) that for $\sigma_h \in \Sigma_h^{\alpha}$ and $\sigma_{h'} \in \Sigma_{h'}^{\alpha}$ with $l^{(s)}(\sigma_h) = \alpha$, $l^{(p)}(\sigma_h) = \beta$ and $l^{(f)}(\sigma_{h'}) = \gamma$, the composite branch label, $L(\sigma_h, \sigma_{h'}) = \alpha G^{s,s} + \gamma G^{f,s} + \beta G^{s,p} + C_{h,h'}^{tr}$ (or the maximum metric of $L(\sigma_h, \sigma_{h'})$) is stored **in the memory with the following index:**

$$\begin{aligned} \mathrm{indx}(\alpha,\beta,\gamma) \triangleq\ & (p_Q(\alpha) + p_{\bar{Q}}(\alpha)F^{(1)}) \circ (p_{\bar{Q}}(\alpha)F^{(2)} + p_R(\gamma) + p_{\bar{R}}(\gamma)E^{(1)}) \\ & \circ (p_{\bar{Q}}(\alpha)F^{(3)} + p_{\bar{R}}(\gamma)E^{(2)} + \beta). \end{aligned} \tag{A.54}$$

[Trellis Construction Procedure Using Isomorphic Parallel Components]

```
For every $\alpha_1 \in \{0,1\}^{\lambda_1}$ {
    Construct $L_{\alpha_1}$, by executing GenerateCBL($\alpha_1$).
    (* Construct isomorphic parallel components. *)
    For every $\alpha_0 \in \{0,1\}^{\lambda_0}$ {
        Let $\alpha$ be an element in $\{0,1\}^{\rho_{h,h'}}$ such that

                $p_Q(\alpha) = \alpha_1 + \alpha_0 F^{(1)}$,     and     $p_{\bar{Q}}(\alpha) = \alpha_0$.

        Construct $\Lambda_\alpha$ by executing Construct$\Lambda(\alpha)$ subprocedure stated below.
    }
}
```

△△

The following subprocedure **Construct$\Lambda(\alpha)$** to construct $\Lambda(\alpha)$ is one to list

$$(l(\sigma_h), l(\sigma_{h'}), \mathrm{indx}(l^{(s)}(\sigma_h), l^{(p)}(\sigma_h), l^{(f)}(\sigma_{h'})))$$

for every state pair $(\sigma_h, \sigma_{h'}) \in \Sigma_h^{\alpha} \times \Sigma_{h'}^{\alpha}$.

Subprocedure Construct$\Lambda(\alpha)$:

```
(* Construct a parallel component $\Lambda_\alpha$. *)
For every $\gamma \in \{0,1\}^{\rho_{h'} - \rho_{h,h'}}$ {
    For every $\beta \in \{0,1\}^{\rho_h - \rho_{h,h'}}$ {
            Output $(l(s_{h,\alpha}(\beta)), l(s_{h',\alpha}(\gamma)), \mathrm{indx}(\alpha,\beta,\gamma))$.
    }
}
```

△△

Note that the labels, $l(s_{h,\alpha}(\beta))$, $l(s_{h',\alpha}(\gamma))$ and $\mathrm{indx}(\alpha,\beta,\gamma)$ are given by (A.29), (A.27) and (A.28), and (A.54), respectively.

REFERENCES

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol.IT-20, pp.284–287, 1974.

[2] A.H. Banihashemi and I.F. Blake, "Trellis complexity and minimal trellis diagrams of lattices," *IEEE Trans. Inform. Theory*, submitted for publication, 1996.

[3] G. Battail, "Pondération des symboles décodés par l'algorithm de Viterbi," *Ann. Télécommun.*, vol. 42, pp.31–38, 1987.

[4] E. Berlekamp, *Algebraic Coding Theory*, Agean Park Press, Laguna Hills, Revised edition, 1984.

[5] Y. Berger and Y. Be'ery, "Bounds on the trellis size of linear block codes," *IEEE Trans. Inform. Theory*, vol.39, pp.203–209, 1993.

[6] Y. Berger and Y. Be'ery, "Soft trellis-based decoder for linear block codes," *IEEE Trans. Inform. Theory*, vol.40, pp.764–773, 1994.

[7] Y. Berger and Y. Be'ery, "Trellis-oriented decomposition and trellis-complexity of composite length cyclic codes," *IEEE Trans. Inform. Theory*, vol.41, pp.1185–1191, 1995.

[8] Y. Berger and Y. Be'ery, "The twisted squaring construction, trellis complexity and generalized weights of BCH and QR codes," *IEEE Trans. Inform. Theory*, vol.42, pp.1817–1827, 1996.

[9] C.Berrou, P.Adde, E.Angui and S.Faudeil, "A low complexity soft-output Viterbi decoder architecture," *in Proc. of ICC*, pp.737–740, 1993.

[10] P.J. Black and T.H. Meng, "A 140-Mb/s, 32-state, radix-4, Viterbi decoder," *IEEE J. Solid-State Circuits*, vol.27, December 1992.

[11] R.E. Blahut, *Theory and Practice of Error Correcting Codes*, Addition-Wesley, Reading MA, 1984.

[12] I.F. Blake and V. Tarokh, "On the trellis complexity of the densest lattice packings in R^n," SIAM J. Discrete Math., vol.9, pp.597–601, 1996.

[13] A.R. Calderbank, "Multilevel codes and multi-stage decoding," *IEEE Trans. Commun.*, vol.37, no.3, pp.222–229, March 1989.

[14] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol.18, no.1, pp.170–181, January 1972.

[15] J.F. Cheng and R.J. McEliece, "Near capacity codes for the Gaussian channel based on low-density generator matrices," in *Proc. 34-th Allerton Conf. on Comm., Control, and Computing*, Monticello, IL., pp.494–503, October 1996.

[16] G.C. Clark and J.B. Cain, *Error Correcting Codes for Digital Communications*, Plenum Press, New York, 1981.

[17] Y. Desaki, T. Fujiwara and T. Kasami, "A Method for Computing the Weight Distribution of a Block Code by Using Its Trellis Diagram," *IEICE Trans. Fundamentals*, vol.E77-A, no.8, pp.1230–1237, August 1994.

[18] S. Dolinar, L. Ekroot, A.B. Kiely, R.J. McEliece, and W. Lin, "The permutation trellis complexity of linear block codes," in *Proc. 32-nd Allerton Conf. on Comm., Control, and Computing*, Monticello, IL., pp.60–74, September 1994.

[19] A. Engelhart, M. Bossert, and J. Maucher, "Heuristic algorithms for ordering a linear block code to reduce the number of nodes of the minimal trellis," *IEEE Trans. Inform. Theory*, submitted for publication, 1996.

[20] J. Feigenbaum, G.D. Forney Jr., B.H. Marcus, R.J. McEliece, and A. Vardy, Special issue on "Codes and Complexity," *IEEE Trans. Inform. Theory*, vol.42, November 1996.

[21] G. Fettweis and H. Meyr, "Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck," *IEEE Trans. Commun.*, vol.37, no.8, pp.785–789, August 1989.

[22] G.D. Forney Jr., *Concatenated Codes*, MIT Press, Cambridge, MA, 1966.

[23] G.D. Forney Jr., "The Viterbi algorithm," *Proc. IEEE*, vol.61, pp.268–278, 1973.

[24] G.D. Forney Jr., "Coset codes II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol.34, pp.1152–1187, 1988.

[25] G.D. Forney Jr., "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol.40, pp.1741–1752, 1994.

[26] G.D. Forney Jr., "Dimension/length profiles and trellis complexity of lattices," *IEEE Trans. Inform. Theory*, vol.40, pp.1753–1772, 1994.

[27] G.D. Forney Jr., "The forward-backward algorithm,"in *Proc. 34-th Allerton Conf. on Comm., Control, and Computing, Control, and Computing*, Monticello, IL., pp.432–446, October 1996

[28] G.D. Forney, Jr. and M.D. Trott, "The dynamics of group codes: state spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inform. Theory*, vol.39, pp.1491–1513, 1993.

[29] G.D. Forney, Jr. and M.D. Trott, "The dynamics of group codes: dual group codes and systems," preprint.

[30] M.P.C. Fossorier and S. Lin, "Coset codes viewed as terminated convolutional codes," *IEEE Trans. Comm.*, vol.44, pp.1096–1106, September 1996.

[31] M.P.C. Fossorier and S. Lin, "Some decomposable codes: the $|a+x|b+x|a+b+x|$ construction," *IEEE Trans. Inform. Theory*, vol.IT-43, pp.1663–1667, September 1997.

[32] M.P.C. Fossorier and S. Lin, "Differential trellis decoding of conventional codes," *IEEE Trans. Inform. Theory*, submitted for publication.

[33] M.P.C. Fossorier, S. Lin, and D. Rhee, "Bit error probability for maximum likelihood decoding of linear block codes," in *Proc. International Symposium on Inform. Theory and its Applications*, Victoria, Canada, pp.606–609, September 1996, and submitted to *IEEE Trans. Inform. Theory*, June 1996.

[34] M.P.C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and Max-Log-MAP decoding," *IEEE Comm. Letters*, submitted for publication, August 1997.

[35] B.J. Frey and F.R. Kschischang, "Probability propagation and iterative decoding," in *Proc. 34-th Allerton Conf. on Comm., Control, and Computing*, Monticello, IL., pp.482–493, October 1996.

[36] T. Fujiwara, T. Kasami, R.H. Morelos-Zaragoza, and S. Lin, "The state complexity of trellis diagrams for a class of generalized concatenated codes," in *Proc. 1994 IEEE International Symposium on Inform. Theory*, Trondheim, Norway, p.471, June/July, 1994.

[37] T. Fujiwara, H. Yamamoto, T. Kasami, and S. Lin, "A trellis-based recursive maximum likelihood decoding algorithm for linear codes," *IEEE Trans. Inform. Theory*, vol.44, to appear, 1998.

[38] P.G. Gulak and T. Kailath, "Locally connected VLSI architectures for the Viterbi algorithms," *IEEE J. Select Areas Commun.*, vol.6, pp.526–637, April 1988.

[39] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Globecom Conference*, Dallas, TX, pp.1680–1686, November 1989.

[40] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol.42, pp.429–445, March 1996.

[41] B. Honary and G. Markarian, *Trellis decoding of block codes: a practical approach*, Kluwer Academic Publishers, Boston, MA, 1997.

[42] G.B. Horn and F.R. Kschischang, "On the intractability of permuting a block code to minimize trellis complexity," *IEEE Trans. Inform. Theory*, vol.42, pp.2042–2048, 1996.

[43] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inform. Theory*, vol.23, no.3, May 1977.

[44] T. Kasami, T. Fujiwara, Y. Desaki, and S. Lin, "On branch labels of parallel components of the L-section minimal trellis diagrams for binary linear block codes," *IEICE Trans. Fundamentals*, vol.E77-A, no.6, pp.1058–1068, June 1994.

[45] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Trans. Inform Theory*, vol.39, no.1, pp.242–243, January 1993.

[46] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inform. Theory*, vol.39, no.3, pp.1057–1064, May 1993.

[47] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On structural complexity of the L-section minimal trellis diagrams for binary linear block codes," *IEICE Trans. Fundamentals*, vol.E76-A, no.9, pp.1411–1421, September 1993.

[48] T. Kasami, T. Koumoto, T. Fujiwara, H. Yamamoto, Y. Desaki and S. Lin. "Low weight subtrellises for binary linear block codes and their

applications" *IEICE Trans. Fundamentals*, vol.E80-A, no.11, pp.2095–2103, November 1997.

[49] T. Kasami, T. Sugita and T. Fujiwara, "The split weight (w_L, w_R) enumeration of Reed-Muller codes for $w_L + w_R < 2d_{\min}$," in *Lecture Notes in Comput. Sci. (Proc. 12th International Symposium, AAECC-12)*, vol.1255, pp.197–211, Springer-Verlag, June 1997.

[50] T. Kasami, H. Tokushige, T. Fujiwara, H. Yamamoto and S. Lin, "A recursive maximum likelihood decoding algorithm for Reed-Muller codes and related codes," *Technical Report of NAIST*, no.NAIST-IS-97003, January 1997.

[51] A.B. Kiely, S. Dolinar, R.J. McEliece, L. Ekroot, and W. Lin, "Trellis decoding complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol.42, pp.1687–1697, 1996.

[52] T. Komura, M. Oka, T. Fujiwara, T. Onoye, T. Kasami, and S. Lin, "VLSI architecture of a recursive maximum likelihood decoding algorithm for a $(64, 35)$ subcode of the $(64, 42)$ Reed-Muller code," in *Proc. International Symposium on Inform. Theory and Its applications*, Victoria, Canada, pp.709–712, September 1996.

[53] A.D. Kot and C. Leung, "On the construction and dimensionality of linear block code trellises," in *Proc. IEEE Int. Symp. Inform. Theory*, San Antonio, TX., p.291, 1993.

[54] F.R. Kschischang, "The combinatorics of block code trellises," in *Proc. Biennial Symp. on Commun.*, Queen's University, Kingston, ON., Canada, May 1994.

[55] F.R. Kschischang, "The trellis structure of maximal fixed-cost codes," *IEEE Trans. Inform. Theory*, vol.42, pp.1828–1838, 1996.

[56] F.R. Kschischang and G.B. Horn, "A heuristic for ordering a linear block code to minimize trellis state complexity," in *Proc. 32-nd Allerton Conf. on Comm., Control, and Computing*, pp.75–84, Monticello, IL., September 1994.

[57] F.R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. Inform. Theory*, vol.41, pp.1924-1937, 1995.

[58] A. Lafourcade and A. Vardy, "Asymptotically good codes have infinite trellis complexity," *IEEE Trans. Inform. Theory*, vol.41, pp.555-559, 1995.

[59] A. Lafourcade and A. Vardy, "Lower bounds on trellis complexity of block codes," *IEEE Trans. Inform. Theory*, vol.41, pp.1938–1954, 1995.

[60] A. Lafourcade and A. Vardy, "Optimal sectionalization of a trellis," *IEEE Trans. Inform. Theory*, vol.42, pp.689–703, 1996.

[61] L. Lin and R.S. Cheng, "Improvements in SOVA-based decoding for turbo codes," *Proc. of ICC*, pp.1473–1478, 1997.

[62] S. Lin and D.J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, NY, 1983.

[63] S. Lin, G.T. Uehara, E. Nakamura, and W.P. Chu, "Circuit design approaches for implementation of a subtrellis IC for a Reed-Muller subcode," *NASA Technical Report* 96–001, February 1996.

[64] C.C. Lu and S.H. Huang, "On bit-level trellis complexity of Reed-Muller codes," *IEEE Trans. Inform. Theory*, vol.41, pp.2061–2064, 1995.

[65] H.H. Ma and J.K. Wolf, "On tail biting convolutional codes," *IEEE Trans. Commun.*, vol.34, pp.104–111, 1986.

[66] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland, Amsterdam, 1977.

[67] J.L. Massey, "Foundation and methods of channel encoding," in *Proc. Int. Conf. Inform. Theory and Systems*, NTG-Fachberichte, Berlin, 1978.

[68] J.L. Massey and M.K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Computers*, vol.COMP-17, pp.330-337, April 1968.

[69] R.J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inform. Theory*, vol.42, pp.1072–1092, 1996.

[70] R.J. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Trans. Inform. Theory*, vol.42, pp.1855–1864, 1996.

[71] A.M. Michelson and D.F. Freeman, "Viterbi decoding of the (63, 57) Hamming codes — Implementation and performance results," *IEEE Trans. Comm.*, vol.43, pp.2653–2656, 1995.

[72] H.T. Moorthy and S. Lin, "On the labeling of minimal trellises for linear block codes," in *Proc. International Symposium on Inform. Theory and its Applications*, Sydney, Australia, vol.1, pp.33–38, November 20–24, 1994.

[73] H.T. Moorthy, S. Lin, and G.T. Uehara, "Good trellises for IC implementation of Viterbi decoders for linear block codes," *IEEE Trans. Comm.*, vol.45, pp.52–63, 1997.

[74] H.T. Moorthy and S. Lin, "On the trellis structure of (64, 40, 8) subcode of the (64, 42, 8) third-order Reed-Muller code," *NASA Technical Report*, 1996.

[75] H.T. Moorthy, S. Lin, and T. Kasami, "Soft-decision decoding of binary linear block codes based on an iterative search algorithm," *IEEE Trans. Inform. Theory*, vol.43, no.3, pp.1030–1040, May 1997.

[76] R.H. Morelos-Zaragoza, T. Fujiwara, T. Kasami, and S. Lin, "Constructions of generalized concatenated codes and their trellis-based decoding complexity," *IEEE Trans. Inform. Theory*, submitted for publication, 1996.

[77] D.J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol.34, pp.1049–1053, 1988.

[78] D.E. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *IEEE Trans. Computers*, vol.3, pp.6–12, 1954.

[79] J.K. Omura, "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory*, vol.IT-15, pp.177–179, 1969.

[80] L.C. Perez, "Coding and modulation for space and satellite communications," Ph.D Thesis, University of Notre Dame, December 1994.

[81] W.W. Peterson and E.J. Weldon, Jr., *Error Correcting Codes*, 2nd edition, MIT Press, Cambridge, MA, 1972.

[82] J.G. Proakis, *Digital Communications*, 3rd edition, McGraw-Hill, New York, 1995.

[83] I.S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IEEE Trans. Inform. Theory*, vol.IT-4, pp.38–49, 1954.

[84] I. Reuven and Y. Be'ery, "Entropy/length profiles and bounds on trellis complexity of nonlinear codes," *IEEE Trans. Inform. Theory*, submitted for publication, 1996.

[85] D. Rhee and S. Lin, "A low complexity and high performance concatenated scheme for high speed satellite communications," *NASA Technical Report*, October 1993.

[86] R.Y. Rubinstein, *Simulated and the Monte Carlo Method*, *New York, Wiley*, 1981.

[87] G. Schnabl and M. Bossert, "Soft decision decoding of Reed-Muller codes and generalized multiple concatenated codes," *IEEE Trans. Inform. Theory*, vol.41, no.1, pp.304–308, January 1995.

[88] P. Schuurman, "A table of state complexity bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol.42, pp.2034–2042, 1996.

[89] V.R. Sidorenko, "The minimal trellis has maximum Euler characteristic $|V| - |E|$," *Problemy Peredachi Informatsii*, to appear.

[90] V.R. Sidorenko, G. Markarian, and B. Honary, "Minimal trellis design for linear block codes based on the Shannon product," *IEEE Trans. Inform. Theory*, vol.42, pp.2048–2053, 1996.

[91] V.R. Sidorenko, I. Martin, and B. Honary, "On separability of nonlinear block codes," *IEEE Trans. Inform. Theory*, submitted for publication, 1996.

[92] G. Solomon and H.C.A. van Tilborg, "A connection between block and convolutional codes," *SIAM J. Appl. Math.*, vol.37, pp.358–369, 1979.

[93] V. Sorokine, F.R. Kschischang, and V. Durand, "Trellis-based decoding of binary linear block codes," in *Lecture Notes in Comput. Sci. (Proc. 3rd Canadian Workshop on Inform. Theory)*, vol.793, pp.270–286, Springer-Verlag, 1994.

[94] D.J. Taipale and M.B. Pursley, "An improvement to generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol.37, no.1, pp.167–172, 1991.

[95] T. Takata, S. Ujita, T. Kasami, and S. Lin, "Multistage decoding of multilevel block M-PSK modulation codes and it performance analysis," *IEEE Trans. Inform. Theory*, vol.39, no.4, pp.1024–1218, 1993.

[96] T. Takata, Y. Yamashita, T. Fujiwara, T. Kasami, and S. Lin, "Suboptimum decoding of decomposable block codes," *IEEE Trans. Inform. Theory*, vol.40, no.5, pp.1392–1405, 1994.

[97] V. Tarokh and I.F. Blake, "Trellis complexity versus the coding gain of lattices I," *IEEE Trans. Inform. Theory*, vol.42, pp.1796–1807, 1996.

[98] V. Tarokh and I.F. Blake, "Trellis complexity versus the coding gain of lattices II," *IEEE Trans. Inform. Theory*, vol.42, pp.1808–1816, 1996.

[99] V. Tarokh and A. Vardy, "Upper bounds on trellis complexity of lattices," *IEEE Trans. Inform. Theory*, vol.43, pp.1294–1300, 1997.

[100] H. Thapar and J. Cioffi, "A block processing method for designing high-speed Viterbi detectors," in *Proc. ICC*, vol.2, pp.1096–1100, June 1989.

[101] H. Thirumoorthy, "Efficient near-optimum decoding algorithms and trellis structure for linear block codes," Ph.D dissertation, Dept. of Elec. Engrg., University of Hawaii at Manoa, November 1996.

[102] A. Vardy and Y. Be'ery, "Maximum-likelihood soft decision decoding of BCH codes," *IEEE Trans. Inform. Theory* vol.40, pp.546–554, 1994.

[103] A. Vardy and F.R. Kschischang, "Proof of a conjecture of McEliece regarding the expansion index of the minimal trellis," *IEEE Trans. Inform. Theory*, vol.42, pp.2027–2033, 1996.

[104] V.V. Vazirani, H. Saran, and B. Sunder Rajan, "An efficient algorithm for constructing minimal trellises for codes over finite abelian groups," *IEEE Trans. Inform. Theory*, vol.42, pp.1839–1854, 1996.

[105] A.J. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol.IT-13, pp.260–269, 1967.

[106] Y.-Y. Wang and C.-C. Lu, "The trellis complexity of equivalent binary (17,9) quadratic residue code is five," in *Proc. IEEE Int. Symp. Inform. Theory*, San Antonio, TX., p.200, 1993.

[107] Y.-Y. Wang and C.-C. Lu, "Theory and algorithm for optimal equivalent codes with absolute trellis size," preprint.

[108] X.A. Wang and S.B. Wicker, "The design and implementation of trellis decoding of some BCH codes," *IEEE Trans. Commun.*, submitted for publication, 1997.

[109] J.K. Wolf, "Efficient maximum-likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol.24, pp.76–80, 1978.

[110] J. Wu, S. Lin, T. Kasami, T. Fujiwara, and T. Takata, "An upper bound on effective error coefficient of two-stage decoding and good two level decompositions of some Reed-Muller codes," *IEEE Trans. Comm.*, vol.42, pp.813–818, 1994.

[111] H. Yamamoto, "Evaluating the block error probability of trellis codes and designing a recursive maximum likelihood decoding algorithm of linear block codes using their trellis structure," Ph.D dissertation, Osaka University, March 1996.

[112] H. Yamamoto, H. Nagano, T. Fujiwara, T. Kasami, and S. Lin, "Recursive MLD algorithm using the detail trellis structure for linear block codes and its average complexity analysis," in *Proc. 1996 International Symposium on Inform. Theory and Its Applications*, Victoria, Canada, pp.704–708, September 1996.

[113] Ø. Ytrehus, "On the trellis complexity of certain binary linear block codes," *IEEE Trans. Inform. Theory*, vol.40, pp.559–560, 1995.

[114] V.A. Zinoviev, "Generalized cascade codes," *Problemy Peredachi Informatsii*, vol.12.1, pp.2–9, 1976.

[115] V.V. Zyablov and V.R. Sidorenko, "Bounds on complexity of trellis decoding of linear block codes," *Problemy Peredachi Informatsii*, vol.29, pp.3–9, 1993 (in Russian).

INDEX